

**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Казанский национальный исследовательский технологический университет»**

**Институт управления, автоматизации и информационных технологий**

**Факультет управления и автоматизации**

**Кафедра автоматизированных систем сбора и обработки информации**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ**

**по дисциплине: «Программирование»**

**на тему: «Создание программы на языке С++»**

Вариант №92

Исполнитель:

студент группы 8191-21

Шайманов Дмитрий Николаевич

Проверил:

доцент Герке А.Р.

Оценка курсовой работы:\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАЗАНЬ, 2020

**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Казанский национальный исследовательский технологический университет»**

**Институт управления, автоматизации и информационных технологий**

**Факультет управления и автоматизации**

**Кафедра автоматизированных систем сбора и обработки информации**

**ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

**по дисциплине «Программирование»**

студенту группы 8191-21

Шайманов Дмитрий Николаевич

1. **Срок представления работы к защите** 2 декабря 2020 г.
2. **Исходная информация к работе**: Вариант №92: Написать программу “напоминалку”. Программа должна уметь загружаться при старте Windows (отключаемая опция), проверять дату и если она совпадает с запланированным событием – выводить на экран окно с сообщением поверх всех окон. После того как пользователь нажмёт кнопку ОК, сообщение закрывается, а программа завершает работу. Если текущая дата не совпадает с запланированным событием, то программа просто завершает работу.
3. **Содержание работы** согласно требованиям по выполнению курсовой работы.

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Герке А.Р.)

**Аннотация**

В курсовой работе преследуется цель создания прикладного программного обеспечения, функционал которого соответствует предлагаемому варианту задания.

Для разработки программного обеспечения используется интегрированная среда разработки программного обеспечения Microsoft Visual Studio. Создание приложения производится с использованием современных технологий программирования на языке C++.

С целью изучения используемой среды разработки, освоения и применения навыков работы со средой на этапе практического создания программного обеспечения, использовалась MSDN - библиотека официальной технической документации для разработчиков под ОС Microsoft Windows.

Пояснительная записка состоит из теоретической и практической частей.

Теоретическая часть включает в себя описания метода решения задачи, обоснование выбор метода, с помощью которого осуществляется решение задачи, его преимущества и недостатки по сравнению с другим методами. Также описываются, какие стандартные функции или функции пользователя будут использоваться в программе, какова будет структура программы.

Практическая часть содержит блок-схему, листинг программы в среде Microsoft Visual Studio, результаты работы программы.

Оглавление

|  |  |
| --- | --- |
| 1. Задание ………………………………………………………………… | 5 |
| 2. Краткое описание метода решения задачи ………………………….. | 6 |
| 3. Подробное описание алгоритма ……………………………………... | 7 |
| 4. Блок-схема программы ……………………………………………….. | 8 |
| 5. Исходный код с комментариями …………………………………….. | 11 |
| 6. Исходный код без комментариев ……………………………………. | 14 |
| 7. Результат работы программы ………………………………………... | 17 |
| 8. Список использованных Интернет-ресурсов и источников ……….. | 20 |

1. **Задание.**

Написать программу “напоминалку”. Программа должна уметь загружаться при старте Windows (отключаемая опция), проверять дату и, если она совпадает с запланированным событием – выводить на экран окно с сообщением поверх всех окон. После того как пользователь нажмёт кнопку ОК, сообщение закрывается, а программа завершает работу. Если текущая дата не совпадает с запланированным событием, то программа просто завершает работу.

1. **Описание метода решения задачи.**

В данной работе я решил использовать язык C#, т.к он наиболее подходит под данное задание, а именно для работы с WindowsForms.

Плюсы данного решения в том, что С# более расположен к WindowsForms, к тому же большинство функций легче использовать именно на этом языке программирования, нежели на С++. В нём я создал форму, откуда буду брать значения даты и сохранять их в параметрах приложения.

Я решил не использовать базу данных SQL, а решил воспользоваться свойствами своего проекта, а именно его параметры. Они позволяют динамически сохранять и извлекать параметры свойств и другие данные для приложения. А так как в данном проекте у меня только одна переменная, которую стоит запоминать, это наиболее эффективный подход. Если поставлена галочка, чтобы приложение запускалось при запуске системы, то я заношу его в реестр приложений для автозапуска.

При повторном включении программа проверяет выбранную дату и системную и, если они совпадают, выдаёт сообщение-напоминание, после чего данные из параметров приложения удаляются, если не совпадают, то программа ничего не делает, если ничего из этого, то запускает первую форму.

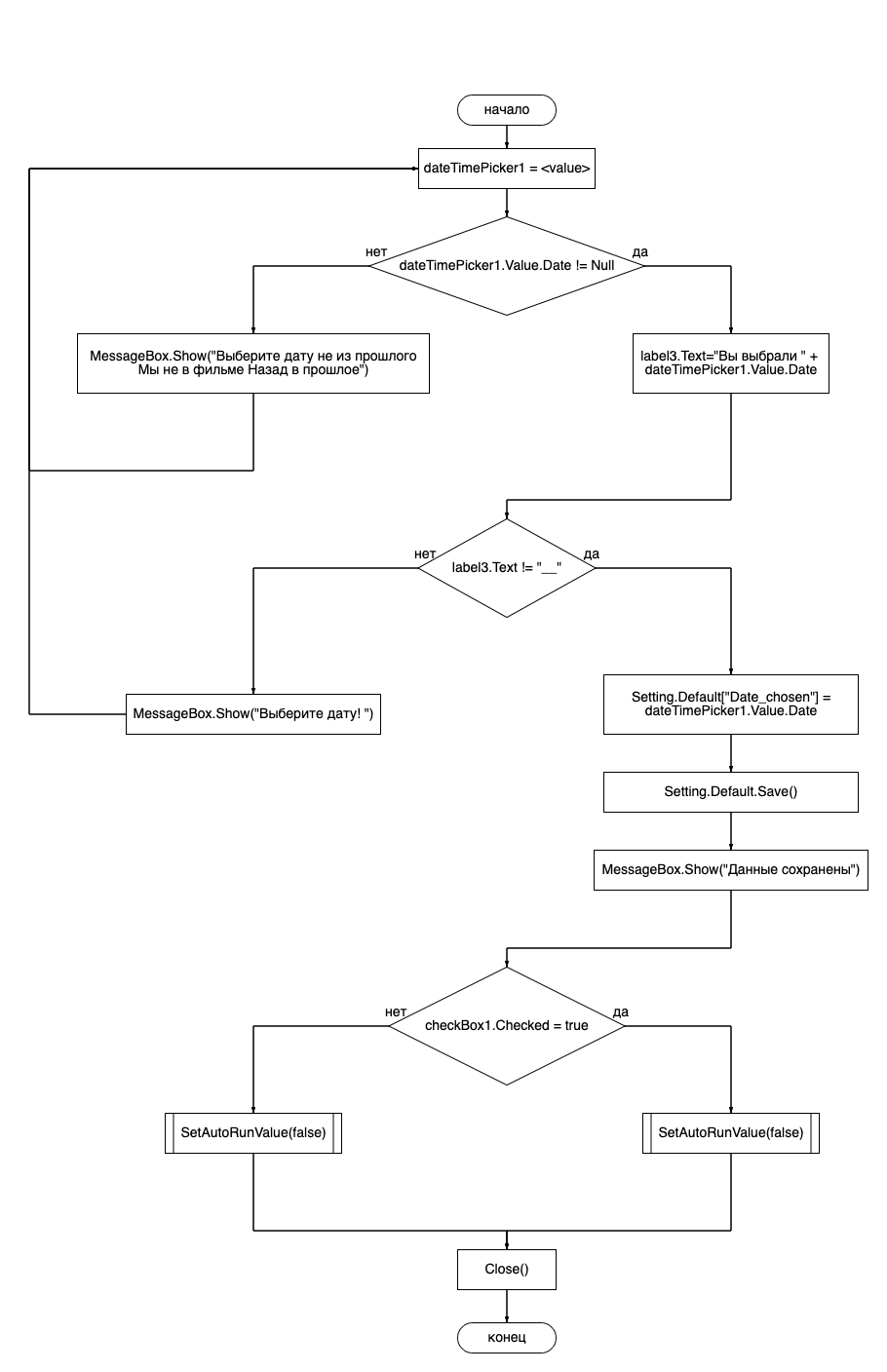
1. **Подробное описание алгоритма.**

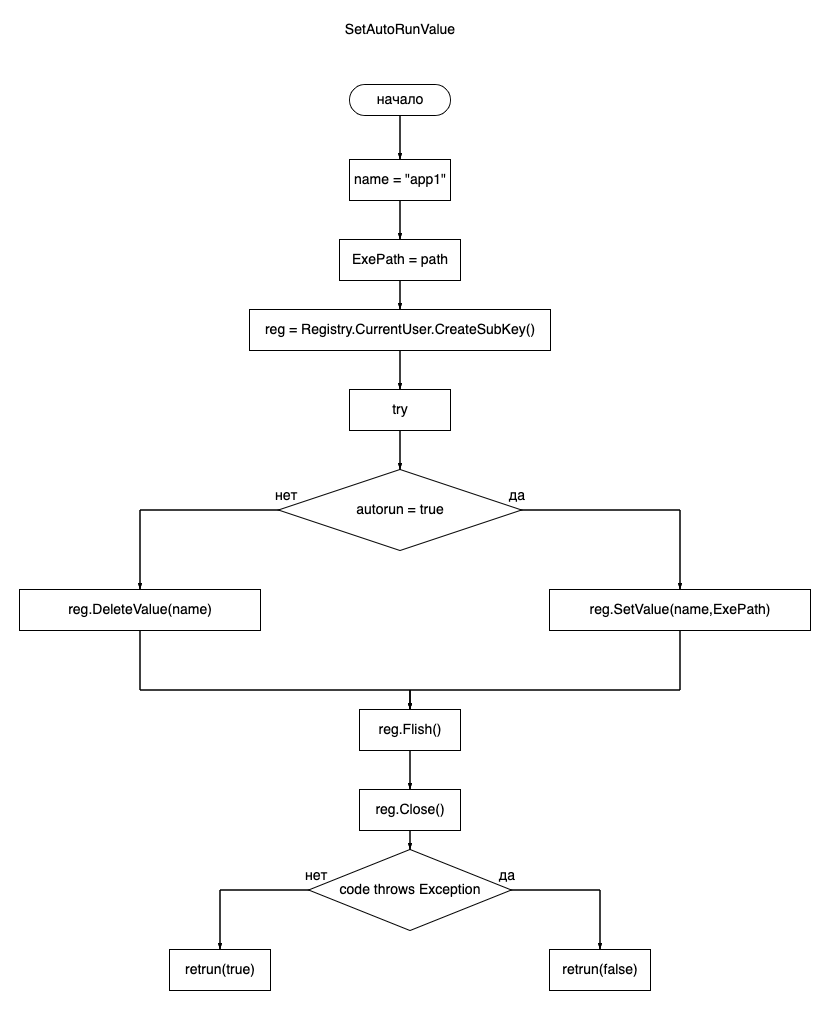
Ещё до запуска программы в файле Program1.csмы прописываем Application.EnableVisualStyles(), для того чтобы отображались визуальные стили приложения, и Application.SetCompatibleTextRenderingDefault(false) для того, чтобы использовать для отрисовки текста класс TextRenderer. Далее, если же дата запланированного события совпадает с сегодняшней датой, то выводим сообщение по верх всех окон, которое напоминает, после чего удаляет дату из параметров приложения, если же дата запланированного события больше сегодняшней, то программа ничего не делает, т. е просто ждёт, если ничего из двух, то запускается приложение.

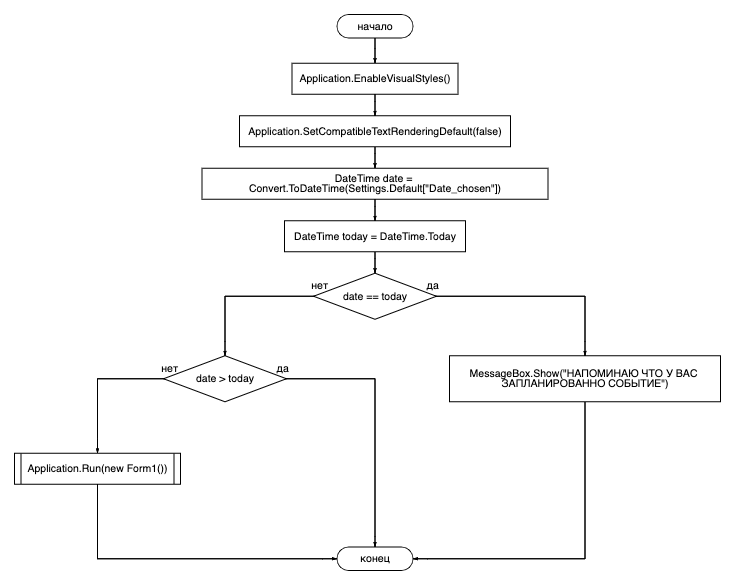
После запуска программы перед нами открывается диалоговое окно, в котором нужно выбрать дату предстоящего события. После того, как мы выберем дату, в окне появится подсказка, которая ещё раз скажет нам какую дату мы выбрали. После этого нам нужно сохранить данные, для этого жмём по кнопке сохранить. Когда мы нажимаем на неё происходит проверка на то ввели ли вы дату или нет: если да, то тогда выбранная дата заноситься в свойства проекта, параметры, и сохраняется там, после этого программа сообщит вам, что данные сохранены; если нет, то появится предупреждение, которое скажет вам, что вы не ввели дату, и попросит сделать это или же, если вы выбрали дату в прошлом, программа попросит сделать выбор даты заново.

Чтобы программа запускалась во время запуска компьютера, нужно поставить галочку в квадратике в поле с одноимённым названием.  
 Если мы поставили галочку, то программа заноситься в реестр программ для автозапуска компьютера. Для этого в SetAutoRunValue мы присваиваем имя нашему приложению, получаем во входных данных путь к файлу и нужна ли автозагрузка или нет. Создаём вложенный ключ для данного пользователя. Если нам автозагрузка нужна, то мы устанавливаем значения ключа – его имя и путь, если нет, то удаляем значение ключа. Записываем в реестр все атрибуты заданного открытого раздела реестра. Закрываем раздел и записываем его на диск. Если же наша функция “словила” Exception, то возвращаем false, если же нет – true.

1. **Блок-схема**







1. **Листинг программы (с комментариями).**

Form1.cs

using System; // для работы с формами

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms; // для работы с формами

using Microsoft.Win32; // для записи в реестр

using App1.Properties; // для записис впараметры приложения даты

using System.Reflection; // для получения пути к фалу

namespace App1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent(); // метод требуемый для конструктора

}

// Форма1

private void Form1\_Load(object sender, EventArgs e)

{

}

// CheckBox

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked == true) // Если автозагрузка нужна

{

SetAutoRunValue(true, Assembly.GetExecutingAssembly().Location); // Assembly.GetExecutingAssembly().Location позволяет получить путь до нашего приложения

}

else // Если автозагрузка не нужна

{

SetAutoRunValue(false, Assembly.GetExecutingAssembly().Location);

}

}

// label1 " Напоминалочка "

private void label1\_Click(object sender, EventArgs e)

{

}

// label 4 Для нормального отображения даты

private void label4\_Click(object sender, EventArgs e)

{

}

// кнопка закрыть

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

// dateTimePicker1

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

label3.Text = "Вы выбрали: " + dateTimePicker1.Value.Date.ToString(); // вывожу выбранную дату в отдельный label3

}

// кнопка Сохранить

private void button1\_Click(object sender, EventArgs e)

{

if (dateTimePicker1.Value.Date < DateTime.Today) // если дата выбрана из прошлого

{

MessageBox.Show("Выберите дату не из прошлого! Мы не в фильме Назад в прошлое "); // выводим окно-сообщение

}

else if (label3.Text != " ") // если label3 не пустой, то сохраняем, если нет, то просим выбрат дату

{

Settings.Default["Date\_chosen"] = dateTimePicker1.Value.Date; // записываем данные в приложение

Settings.Default.Save(); // сохраняем данные в приложении

MessageBox.Show("Данные сохранены!"); // выводим окно-сообщение

}

else // если ничего не введено

{

MessageBox.Show("Выберите дату!"); // выводим окно-сообщение

}

}

// функция для добавления в реестр автозапуска

// переменная autorun отвечает за то, нужна ли функция автозапуска

// path - это путь нашего приложения

private bool SetAutoRunValue(bool autorun, string path)

{

const string name = "app1"; // присваем будущее имя нашему процессу в реестре автозагрузок

string ExePath = path; // присваиваем путь

RegistryKey reg; // создаём переменную reg класса RegistryKey для работы с этим классом

reg = Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\Windows\\CurrentVersion\\Run\\");

// создаём вложенный ключ

// try/catch для того тобы словить Exception

try

{

if (autorun) // если автозагрузка нужна,то устанавливаем значния ключа: его имя и путь к нашему приложению

{

reg.SetValue(name, ExePath);

}

else // если не надо, то просто удаляем вложенный ключ

{

reg.DeleteValue(name);

}

reg.Flush(); // записываем в реестр все атрибуты заданного открытого раздела реестра

reg.Close(); // если содержимое было изменено, закрываем его и записываем на диск

}

catch (Exception)

{

return false;

}

return true;

}

}

}

Program.cs

using System; // для работы с формами

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms; // для работы с формами

using App1.Properties; // чтобы достать выбранную дату из приложения

namespace App1

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles(); // включает визуальный стили приложения

Application.SetCompatibleTextRenderingDefault(false); //для того, чтобы использовать для отрисовки текста класс TextRenderer

DateTime date = Convert.ToDateTime(Settings.Default["Date\_chosen"]); // берём из памяти приложения выбранную дату и конвертируем её

// при помощи метода класса Convert.ToDateTime() из object в DateTime

DateTime today = DateTime.Today; // используя свойства структуры, получаем сегодняшную дату

if (date == today) // если даты равны, то выводоим напоминание

{

MessageBox.Show("НАПОМИНАЮ ЧТО У ВАС ЗАПЛАНИРОВАННО СОБЫТИЕ","НАПОМИНАНИЕ", MessageBoxButtons.OK, MessageBoxIcon.Asterisk, MessageBoxDefaultButton.Button1, MessageBoxOptions.ServiceNotification);

// первый аргумент сообщение, второй название окна, третий какая кнопка, четвёртый картинка, определем кнопку, окно сооющения

Settings.Default.Reset(); // Удаляем данные из памяти приложения

}

else if(date > today)

{

// если выбранная дата больше, чем сегодняшняя, то ничего не делаем

}

else // запуск приложения

{

Application.Run(new Form1());

}

}

}

}

1. **Листинг программы (без комментариев).**

Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using Microsoft.Win32;

using App1.Properties;

using System.Reflection;

namespace App1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked == true)

{

SetAutoRunValue(true, Assembly.GetExecutingAssembly().Location);

}

else

{

SetAutoRunValue(false, Assembly.GetExecutingAssembly().Location);

}

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label4\_Click(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

Close();

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

label3.Text = "Вы выбрали: " + dateTimePicker1.Value.Date.ToString();

}

private void button1\_Click(object sender, EventArgs e)

{

if (dateTimePicker1.Value.Date < DateTime.Today)

{

MessageBox.Show("Выберите дату не из прошлого! Мы не в фильме Назад в прошлое ");

}

else if (label3.Text != " ")

{

Settings.Default["Date\_chosen"] = dateTimePicker1.Value.Date;

Settings.Default.Save();

MessageBox.Show("Данные сохранены!");

}

else

{

MessageBox.Show("Выберите дату!");

}

}

private bool SetAutoRunValue(bool autorun, string path)

{

const string name = "app1";

string ExePath = path;

RegistryKey reg;

reg = Registry.CurrentUser.CreateSubKey("Software\\Microsoft\\Windows\\CurrentVersion\\Run\\");

try

{

if (autorun)

{

reg.SetValue(name, ExePath);

}

else

{

reg.DeleteValue(name);

}

reg.Flush();

reg.Close();

}

catch (Exception)

{

return false;

}

return true;

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

using App1.Properties;

namespace App1

{

static class Program

{

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

DateTime date = Convert.ToDateTime(Settings.Default["Date\_chosen"]);

DateTime today = DateTime.Today;

if (date == today)

{

MessageBox.Show("НАПОМИНАЮ ЧТО У ВАС ЗАПЛАНИРОВАННО СОБЫТИЕ","НАПОМИНАНИЕ", MessageBoxButtons.OK, MessageBoxIcon.Asterisk, MessageBoxDefaultButton.Button1, MessageBoxOptions.ServiceNotification);

Settings.Default.Reset();

}

else if(date > today)

{

}

else

{

Application.Run(new Form1());

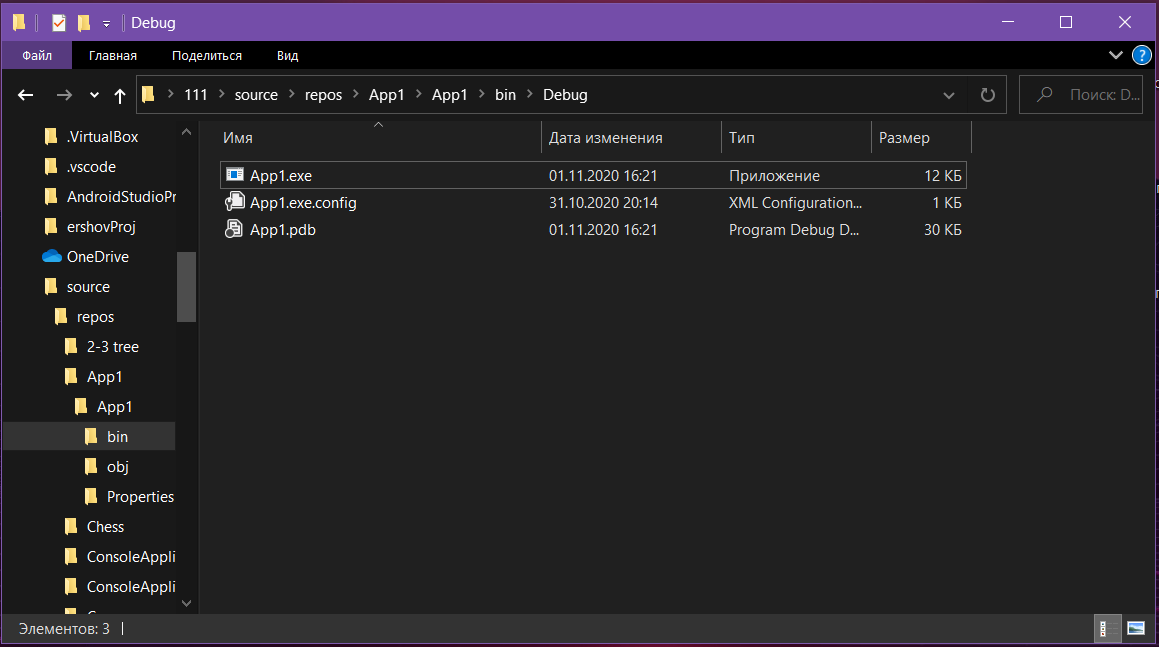
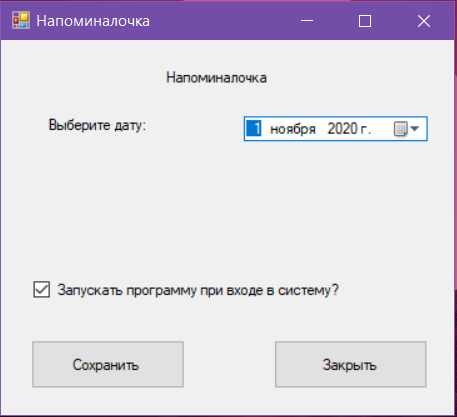
}

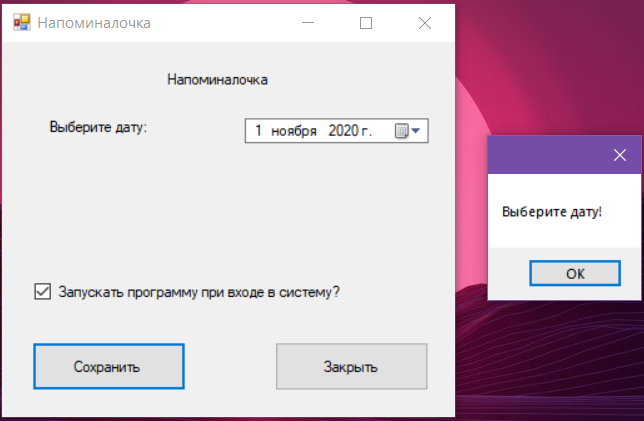
}

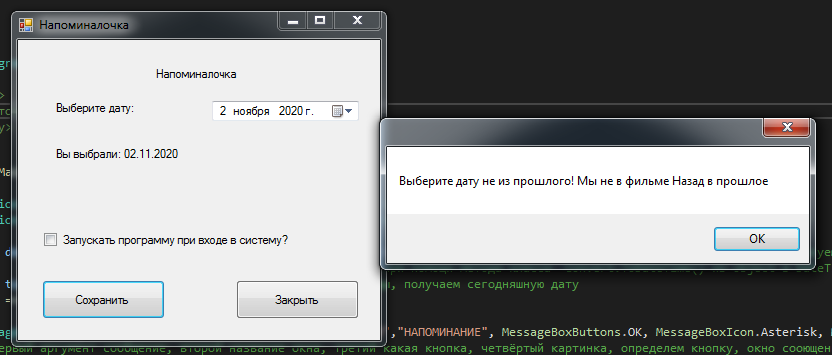
}

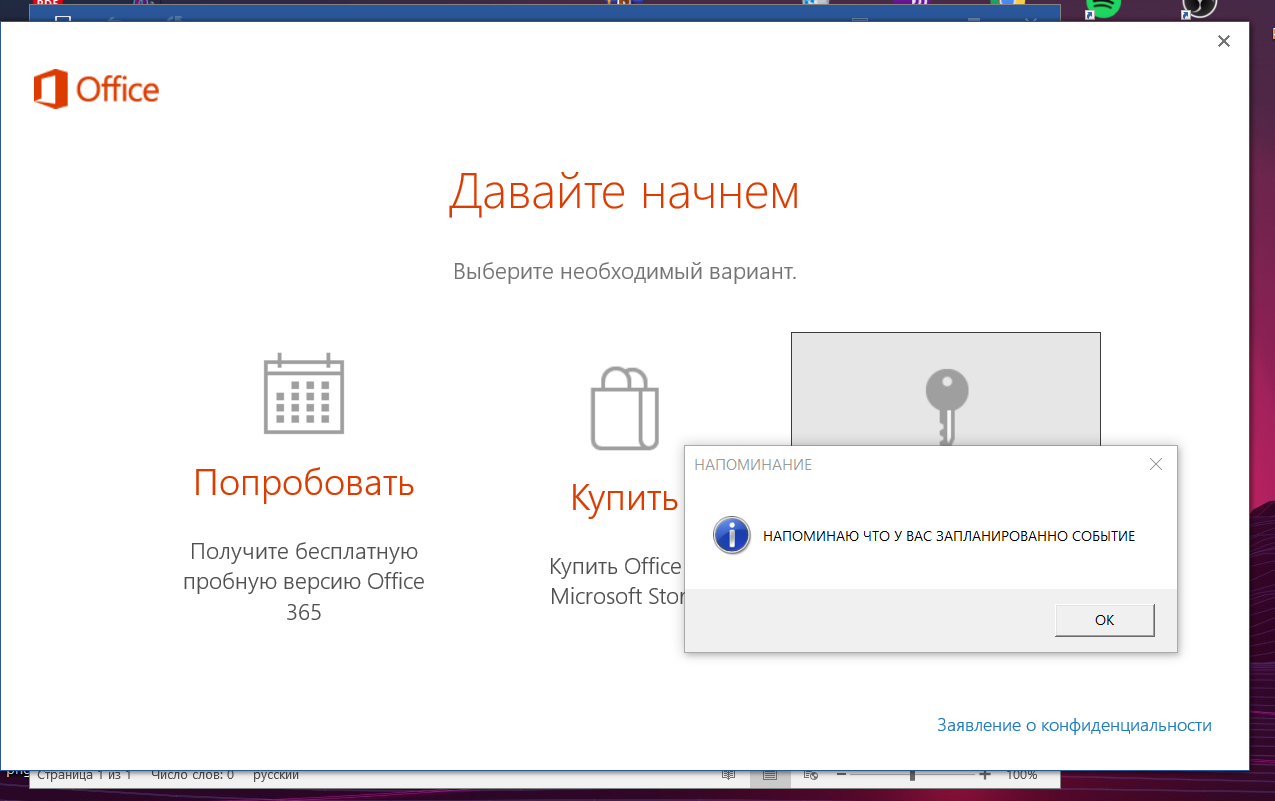
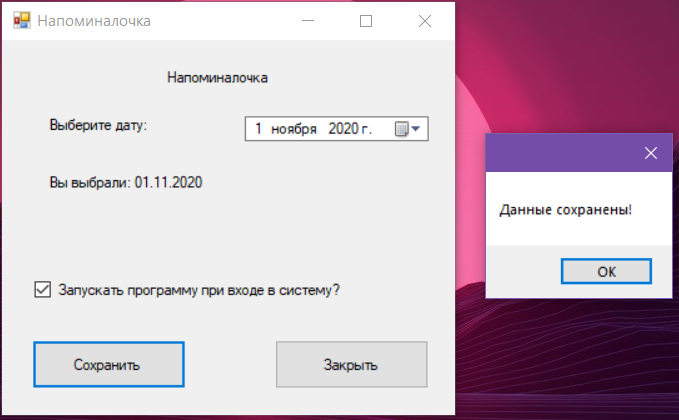
}

1. **Результат работы программы.**









1. **Список литературы**

1.https://docs.microsoft.com/ru-ru/dotnet/api/system?view=netcore-3.1